# Tutorial and Practical 1 – The Basics of C

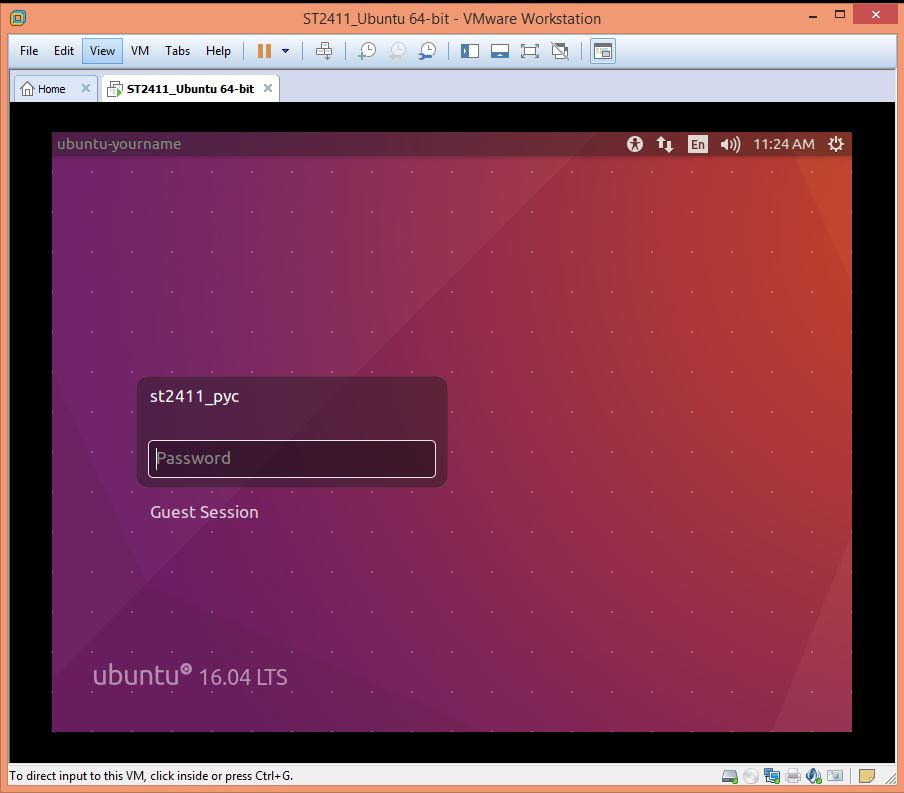
By the end of the lesson, you should be able to

* Compile a C program
* Write a basic C program
* Use data types in C
* Use the printf statement to display and format data
* Use the scanf statement to get user input
* Use simple C functions

Note:

Obtain the VM image : "ST2411\_Ubuntu 64-bit" from your tutor.

You are also advised to use it with VMWare Workstation 12 Pro 12.5.5 or later.

The password of the default user is 1qwer$#@!

All coding for assignments MUST be able to be compiled and run on the class assigned Ubuntu image. Assignments that cannot be compiled and run on this image will automatically receive a failing grade.

You can use the sublime text editor or any other editor (e.g. gedit or vi) as long as your code can be compiled and run on the class assigned image.

Task 1

Login to your Ubuntu VM.

Open a new Terminal.

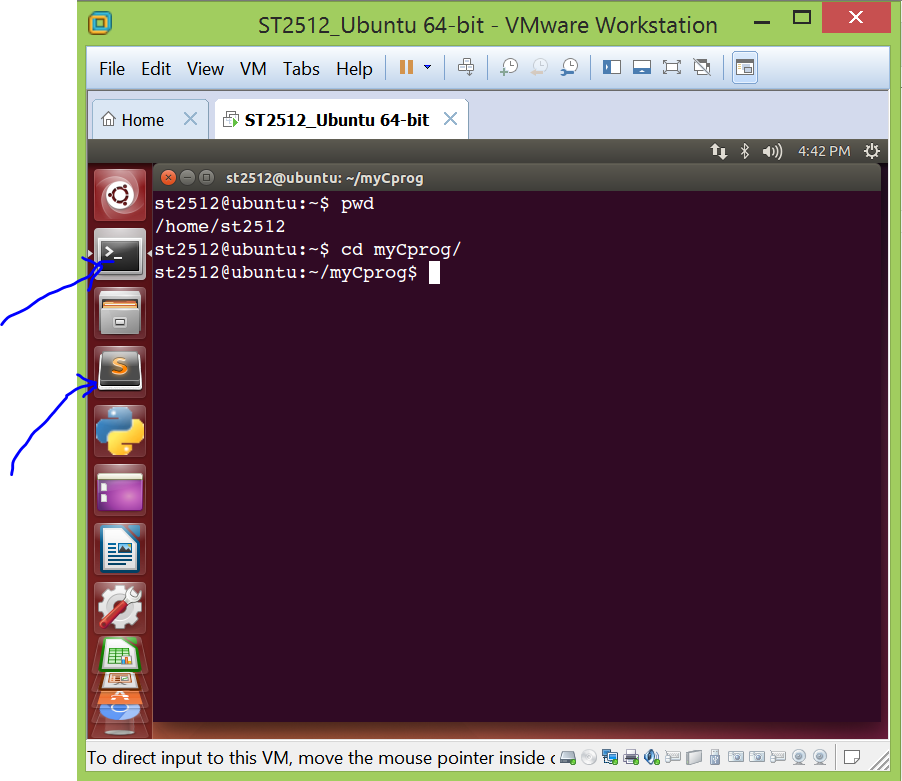
Change current directory to myCprog folder

Useful shell commands :

pwd ( print current working directory)

cd (change directory)

ls (list the content of the current directory



Terminal

File Manager

Sublime Text

Refer to Lecture 1 slide and use a text editor

Create a C Program that prints "This is my first C Hello World".

Save your C program in the file, Hello.c .

To compile, you can use

* cc –o test Hello.c

Try to run the compiled program with

* ./test

And

test

Which one of the above can work? If one of them cannot work, can you explain what the cause of that is?

[hint: try to search the usage for a linux command 'which', it may help you to explain]

[ Try this : Remove the #include <stdio.h> directive from your Program source. Can you compile and execute your program in this case ?]

Task 2

Create a C Program that prints out your name.

Task 3

Compile the previous program using the –o switch and produce an executable file with a name **of your choice**.

Write down the complete compilation command to compile your program created in Task 2 to an executable file of your choice

|  |
| --- |
|  |

Task 4

- Locate the following codes,p1\_task4.c and at your myCprog folder.

#include <stdio.h>

/\* Source File name – p1\_task4.c \*/

int main() {

char \* msg = "Hello World!\n";

printf("%s", msg);

}

Using the following command at the prompt.

$cc p1task4.c

[What is the name of the executable file that the above compilation has generated? ]

Recompile the programming using

cc –Wall p1task4.c

What difference was there in the first compilation and the second compilation? Why? (Hint: use the man cc or google for the answer)

|  |
| --- |
|  |

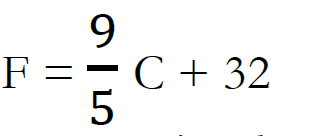
When do you think you should use the –Wall switch?

|  |
| --- |
|  |

*Task 5*

Write a C Program that converts degrees Celsius to degrees

Fahrenheit. The formula for conversion is



Do a sanity check. Ensure that the results are correct for a set of numbers including negative, zero and positive values.

e.g. check a few values that are within the range from -100 Celsius to +100 Celsius

[You may hardcoded the input values in the source code]

*Task 6*

Based on the given p1t6v1.c, modify it to prompt the user to enter an input of a temperature reading in Celsius. The program will then display appropriate messages with the temperature reading in Fahrenheit.

Here is a Sample run of Task 6 solution (User Input is highlighted in Red color)

st2411@ubuntu:~/myCprog$ ./p1t6v1

Please Enter a degree in Celsius=> 0

The eqvalent degree in Fahrenheit is 32.00

st2411@ubuntu:~/myCprog$ ./p1t6v1

Please Enter a degree in Celsius=> 37

The eqvalent degree in Fahrenheit is 98.60

st2411@ubuntu:~/myCprog$ ./p1t6v1

Please Enter a degree in Celsius=> -40

The eqvalent degree in Fahrenheit is -40.00

st2411@ubuntu:~/myCprog$ ./p1t6v1

Please Enter a degree in Celsius=> -55.5

The eqvalent degree in Fahrenheit is -67.90

*Task 7*

Write a C Program that gets a char as an input and displays the ascii value of the char. Use the getchar function.

How many characters can the getchar function accept?

*Task 8*

Copy, compile and run this code.

**/\* Playing with printf formats**

**\* printf.c**

**\* Oct 30 2007**

**\* Chris Szalwinski**

**\* Modified by Karl \*/**

**#include <stdio.h>**

**int main() {**

**/\* integers \*/**

**printf("\n\* ints \*\n");**

**printf("00000000011\n");**

**printf("12345678901\n");**

**printf("------------------------\n");**

**printf("%d|<-- %%d\n",4321);**

**printf("%10d|<-- %%10d\n",4321);**

**printf("%010d|<-- %%010d\n",4321);**

**printf("%-10d|<-- %%-10d\n",4321);**

**/\* floats \*/**

**printf("\n\* floats \*\n");**

**printf("00000000011\n");**

**printf("12345678901\n");**

**printf("------------------------\n");**

**printf("%f|<-- %%f\n",4321.9876546);**

**/\* doubles \*/**

**printf("\n\* doubles \*\n");**

**printf("00000000011\n");**

**printf("12345678901\n");**

**printf("------------------------\n");**

**printf("%lf|<-- %%lf\n",4321.9876546);**

**printf("%10.3lf|<-- %%10.3lf\n",4321.9876546);**

**printf("%010.3lf|<-- %%010.3lf\n",4321.9876546);**

**printf("%+10.3lf|<-- %%+10.3lf\n",4321.9876546);**

**printf("%-10.3lf|<-- %%-10.3lf\n",4321.9876546);**

**/\* characters \*/**

**printf("\n\* chars \*\n");**

**printf("00000000011\n");**

**printf("12345678901\n");**

**printf("------------------------\n");**

**printf("%c|<-- %%c\n",'d');**

**printf("%d|<-- %%d\n",'d');**

**printf("%o|<-- %%o\n",'d');**

**printf("%x|<-- %%x\n",'d');**

**return 0;**

**}**

Study the program output and answer the following question:

What do the different "place holder characters" values represent/do?

[ie. %c, %d, %o, %x, %f, %lf]

*Task 9*

Modify your program from **task 7** to display the user input character in terms of Actual Character, Hex Value and Decimal Value.

For example , if the user enter '5' follow by an enter key. The program shall display the following :

Character Hex Value Decimal Value

5 35 53

Note the column alignments (and/or) spaces between the columns

*Task 10*

Trying out the following C Program :

|  |
| --- |
| /\* sample C program to try out scanf() \*/  #include <stdio.h>  int main() {  int x ; float y ; char z ; char s[20] ;  scanf("%d", &x);  scanf("%f", &y);  scanf("%c", &z);  scanf("%s",&s[0]); /\* read in one word \*/  /\* echo back all input \*/  printf("%d %f %c %s \n",x, y, z, s);  } |

[Figure out how to make the program displaying the output like the following :]

10 0.300000 A 67

Task 11

Write a C Program to prompt for the area of a square, then it will calculate and display the length of one side, the perimeter of the square and project the volume and surface area if the square was one of the cube surfaces. Your output must show up to 2 decimal places. [Hint: you may use sqrt and pow functions in your calculation) A sample run:

Please Enter the area of a square => 25

Length of one side = 5.00

Perimeter of the square = 20.00

Volume of the cube = 125.00

Surface area = 150.00

# ~ End ~